网络科技信息监测中富文档识别与信息提取技术研究*

张敏[1] 刘建华[1,2] 谢靖[1]

[1]（中国科学院文献情报中心 北京 100190）

[2]（中国科学院大学 北京 100190）

摘要 本论文围绕富文档载体类型的鉴别、元数据的提取等开展相应的实际应用探索。笔者通过开源工具 PDFBox 以及 Tika 对不同类型的富文档元数据及正文内容进行提取，取得了很好的实际效果，为科研人员提供了大量的有学术价值的情报资源。但是由于开源工具的局限性以及富文档特殊的文档结构，导致提取出来的元数据及正文内容准确率欠缺完美，笔者后续将对此进行研究并完善改进。

关键词 富文档 元数据 类型识别

Identification and Information Extraction of Rich Documents for Web Scientific Information Monitoring

Zhang min[1] Zhang Zhixiong[1] Liu Jianhua[1,2] Xie Jing[1]

1(National Science Library, Chinese Academy of Sciences, Beijing 100190, China)

2(University of Chinese Academy of Sciences,Beijing 100190, China)

Abstract This paper focuses on the practical application of the identification of the rich document carrier, the extraction of metadata and the content of the text,and so on. Through the open source tools, such as PDFBox and Tika, the author has provided a lot of valuable information resources for the scientific research personnel, which has obtained good actual effect. But due to the tool's limitation and the special document structure of rich document, the accuracy rate about the extracted metadata and the content of the text is not perfect.Thus,the author will improve it through the next research.

Keywords Rich Documents、Metadata、Identification of the rich document carrier

随着网络传播方式的广泛普及，越来越多的管理机构、科研机构在通过正式交流渠道（如期刊、图书等媒介）发布研究成果的同时，他们也选择通过 Internet 这类非正式交流平台发布、共享相关的科研新闻、重大成果或研究报告、年度预算等，这些内容除了以 html 形式存在，更多的以 PDF、DOC、PPT 等载体形式存在。后述的几种载体形式文本中除了正文内容外，还包含以标准化的方法对不同文本格式的属性，如加粗、字体以及文档格式等进行编码的信息[1]。参照 OpenDocument 的概念，笔者将其定义为富文档文件。这些富文档文件往往比 html 一类的平文档包含了更为丰富、相对正式的情报知识，是情报分析人员、科研人

* 本文系中国科学院文献情报能力建设专项“网络科技信息自动监测系统三期建设”（课题编号：院 1509）的研究成果之一

员特别关注的重要对象。对基于自动信息采集开展的科技动态监测工作而言，在海量的动态信息采集过程中准确识别筛选出这些富文档的载体类型，充分利用富文本包含的丰富描述内容的信息，如文档中色彩、字形和字体变化对文档内容的显著突出作用[2]，获取其相应的描述元数据，如标题、发布时间、摘要等，并从其正文内容中识别出其中包含的重要科技对象（人物、机构、会议、战略计划、法案等）及科技主题，对进一步开展重要科技信息计算、科技资源间关联关系分析等有重要影响。本论文即围绕富文档载体类型的鉴别、元数据的提取等开展相应的应用探索。

1 富文档监测与识别的相关研究

目前，专门针对多种类型的网络富文档文件进行载体类型的准确识别及元数据信息提取的研究不是太多，其中多数研究围绕着文档物理层次的表达[3]、OCR 句法识别及纠正[4]、基于富文档特征的逻辑结构探测[5]、PDF 文件信息的抽取与分析开展[11]、PDF 科技论文语义元数据的自动抽取研究[12]、基于 PDFBox 抽取 PDF 格式的学术论文[13]等，这些研究对于笔者在提取富文档的相关元数据信息时具有相应的参考意义，如通过分析 PDF 文件的结构提取出文本及其相关的字体、字号和换行等文本信息；采用基于格式的定位方法抽取富文档中的元数据信息；采用开源函数库 PDFBox 对以富文档类型为载体的论文进行标题、发表时间、摘要等元数据的提取；借助富文档文件中字体的变化及文字出现的逻辑位置确定其相应的标题、摘要信息。在开展富文档文件的载体类型时，各类协议标准可以作为识别的重要依据。

2 富文档识别与框架

上文中已经说过，本研究中富文档监测与识别任务包括两个方面，一是富文档文件载体类型的准确识别，第二是富文档文件各类元数据信息提取。因此在该任务中，笔者分别针对两个子任务设计了相应的识别方法。考虑到在后期的实现方法中，富文档文件的载体类型将对其元数据的提取产生重要的影响，如工具的选择等，笔者设定了如下的富文档监测与识别框架（图 1）。

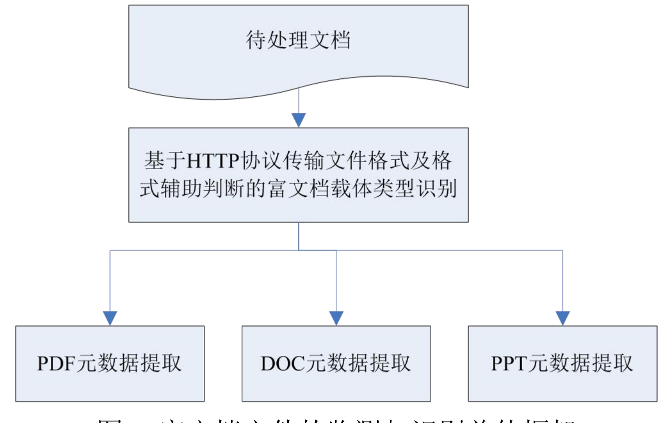


图 1 富文档文件的监测与识别总体框架

整个流程中，首先基于HTTP协议传输文件格式及文件名后缀的辅助识别，确定待处理文档的载体类型，然后依据不同的载体类型选择相应的元数据解析器，其中DOC及PPT由于统一采用了微软的文档标准，因此在某些元数据的解析方面方法相同，但由于其表现特征上的不同，在具体的实现中还有所区别。这将在下文中有详细的说明。

3 富文档监测与识别方法

依据上文设定的总体识别框架，笔者将进一步从富文档文件载体类型的识别元数据信息提取两个方面详细阐述富文档的监测与识别方法。

3.1 富文档载体类型的识别方法

在信息抓取的过程中，文档类型众多、文档类型表示方式多样化、网站的传递方式不是完全的标准化，这些问题对富文档的识别造成一定的影响。使用单一的标准判定方法，会造成一定的误判。因此如何综合各类富文档识别的技术手段，将多种方式有效的组合，提高富文档识别的准确度，是研究的关键。经过比较研究，笔者选择主要基于HTTP协议传输文件格式和文件名后缀识别两种方式来获取富文档文件的载体类型信息。

（1）基于HTTP协议传输文件格式的类型判定

HTTP协议传输文件格式以MIME的标准类型名命名，解析协议头的Content-Type可以获取MIME类型名称，根据MIME多媒体文档类型标准，映射查询得到识别正确的文档类型（RFC-2046 MIME Part 2: Media Types [6]多媒体文件类型规定设定的常见类型名称映射见表1）。以图2为例，这是W3C首页的HTTP头的内容，其中Content-Type包含的MIME类型名称为text/html，根据MIME类型映射，该页面所属文件类型为Html文档。

```
Date=Sat, 11 Dec 2010 13:27:27 GMT
Server=Apache/2
Content-Location=Home.html
Vary=negotiate,accept
TCN=choice
Last-Modified=Sat, 11 Dec 2010 06:07:09 GMT
ETag="7bc1-4971c47f59d40;89-3f26bd17a2f00"
Accept-Ranges=bytes
Content-Length=31681
Cache-Control=max-age=600
Expires=Sat, 11 Dec 2010 13:37:27 GMT
P3P=policyref="http://www.w3.org/2001/05/P3P/p3p.xml"
Connection=close
Content-Type=text/html; charset=utf-8
```

图2 HTTP协议头解析和MIME判断

类型/子类型	扩展名
application/msword	doc
application/pdf	pdf
application/rtf	rtf
application/vnd.ms-excel	xls
application/vnd.ms-outlook	msg
application/vnd.ms-powerpoint	ppt
application/vnd.ms-works	wks
text/html	html
text/plain	txt
text/richtext	rtx

表 1 常见类型名称映射

尽管 MIME 格式识别的文档类型准确性最高，但其识别结果仍然具有一定的二义性。在有二义性发生的情况下，就需要借助于其他方法来进行辅助判定。如需要补充文件名和文件后缀详细指定，在 MIME 映射之后，增加文件名后缀识别，可提高文档格式识别的准确性。但并不是所有网站都采用标准 MIME 类型名指定文档类型，这样就会造成 MIME 类型获取失败，或者无法找到对应文件类型。因此收集其它非标准类型网站的文档类型标识方式，也是富文档识别方式之一。另外，还有某些网站直接采用 URL 后缀的方式或 URL 中嵌入格式类型的方式指定文件格式，这就需要分析 URL，将其映射到相应格式的文件。

（2）文件名后缀判断

MIME 类型名称规定了多媒体文件打开的程序，但是对于详细的文件格式仍有一定的差异。如 application/vnd.ms-powerpoint，规定的该文件应该使用 ms-powerpoint 程序打开，而 ms-powerpoint 可以打开的程序包括 ppt，pps 等多个不同后缀类型的文件。因此要使用补充文件名称，根据文件名称的后缀进行详细文件格式的判断。

Content-Disposition=attachment; filename=A1002.ppt

在补充 HTTP 协议头的 Content-Disposition 内容中，包含了建议保存的文件名称，在 filename 参数指定了文件的完整名称和文件后缀名，可以以文件后缀名为根据判读文件的精确的文件类型。

（3）URL 判断

有一部分网页已经在 URL 中或后缀上，直接标明了文件类型。如 apache 的开源软件 httpclient，其下载地址为：http://labs.renren.com/apache-mirror//httpcomponents/httpclient/binary/httpcomponents-client-4.0.3-bin.zip，其后缀为 zip 压缩文件格式，亦可以作为文件类型判断的依据之一。又如布鲁金斯学会的某个情报产品的图片，其链接地址为 http://www.brookings.edu/~/media/research/images/c/cp%20ct/crowd001_16x9.jpg?w=120，该地址中“.jpg”标明了其图片格式类型。但是，有些 URL 的隐藏载体类型并不在 URL 后缀中，如果依然采取判断 URL 字段后缀的方法肯定没法有效判断其载体类型。例如：有的 URL 地址中隐含了 format=pdf，针对此类 URL 我们也可以得出此 URL 的载体类型是 PDF 格式。

综合上述三种方式的分析，笔者充分融合了这三种方式的优缺点，对其进行排序组合，形成文档类型判断流程（图 3），基于这一组合的流程，笔者可以准确实现对采集下来的富文档文件的载体类型判别。

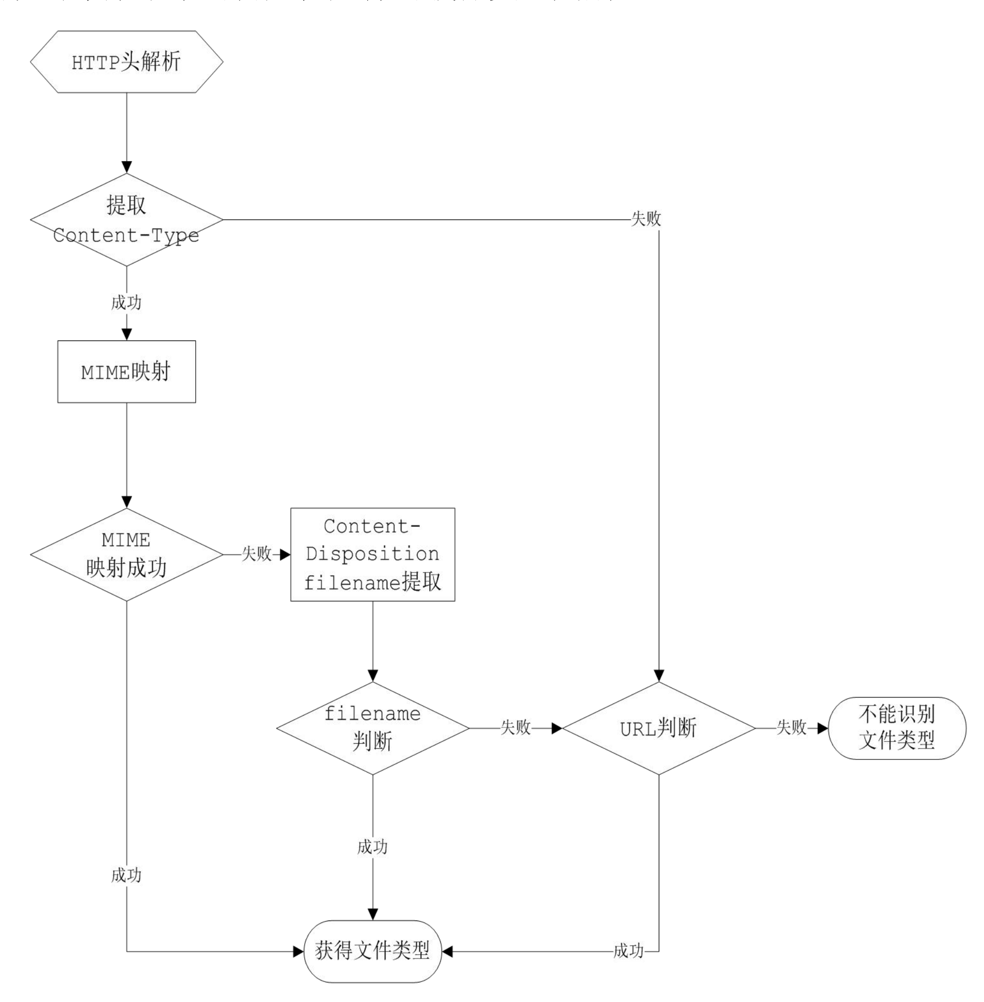


图 3 富文档文件载体类型识别流程

另外在实际的监测资源采集过程中，碰到前三种方法均无法判断的情况，针对此种情况，目前笔者采用了在可配置文件中进行特殊配置的方法来定义其载体类型，如在配置文件中定义：http://rspb.royalsocietypublishing.org/highwire/powerpoint/65580、http://www.avc-mr.com/research/r1/p1/2015 等，碰到此类 URL 或者以此为前缀的 URL，系统自动判断其载体类型为 PDF 文档。

3.2 富文档元数据提取方法

确定了富文档格式类型之后，需要进一步获取这些富文档文件的元数据信息。对监测项目而言，PPT、DOC、PPT、XLS 这几类富文档是关注的重点，因此，笔者重点围绕这几类富文档探索了相关的元数据提取方法。受篇幅的限制，本文中将集中于文档发布时间、标题、正文内容的提取几个方面。

（1） 文档发布时间的提取。

通过大量的统计分析，笔者发现，在富文档的二进制流中往往包含了制作时间、制作软件等丰富的技术元数据，这些信息较好地表现了文档的创建时间、修改时间。因此，笔者针对文档的发布时间的提取主要通过解析其二进制流中属性部分的相关技术信息。以 PDF 文档为例，笔者判定其发布时间的主要方法如图 4 所示，在获取 PDF 的技术信息时，笔者先判定其是否存在更新或者修订的时间信息，若存在，以此为依据作为 PDF 的最终发布时间，若无，则取其创建时间。在实际的应用中，通过这种方法，笔者可以准确解析出 95%以上的富文档文件的发布时间。

```
if ((null != pdfInfo.getModificationDate())
        && (!"".equals(pdfInfo.getModificationDate()))) {
        Calendar cale = pdfInfo.getModificationDate();
        taskStr = tranferTime(cale.getTime());
} else if ((null != pdfInfo.getCreationDate())
        && (!"".equals(pdfInfo.getCreationDate()))) {
        Calendar cale1 = pdfInfo.getCreationDate();
        taskStr = tranferTime(cale1.getTime());
}

if (null != taskStr && (!"".equals(taskStr))
        && (!"".equals(taskStr.trim()))) {
        Timestamp taskTime = Timestamp.valueOf(taskStr);
if (taskTime.before(cadiTime)) {
        stamp = taskTime;
}
}
```

图 4 判定 PDF 发布时间的主要流程判断

（2）文档标题的提取

在对大量富文档文件进行特征分析的过程中，笔者发现富文档文件的标题往往会出现在以下几个位置：链入页面上的锚点文字；url 字段中标出的标题内

容；文本属性中标题属性值；正文内容中顶部最大字体值。而在特征表现上，富文档的标题往往出现在内容页的顶端；往往是内容中最大的字体或者在颜色、字形上与其它内容有所区别；往往不会少于 3 个分词，也不会超过 30 个分词（包括副标题）；往往除介词、连接词外，其它单词首字母为大写或全为大写；往往在标题中出现的词或词组会出现在正文中。

基于这几点分析结果，笔者设计了如下的处理流程：

（1） 从锚点文字中获取文档标题

在监测采集过程中，通过记录链入、链出页面的关系以及相应的锚点文字信息，从中取得文档标题；

（2） 从页面 url 字段中获取文档标题

通过分析 url 字段，利用相关函数取得最后一个字符“/”后的字段，并替换截取后字段中的非字符信息即为文档标题。

（3）从文本属性中获取文档标题。针对这一类的信息，可以分别利用 Apache PDFBox[8]和 Apache POI[9]两个开源工具进行相应的处理和获取。但是这一来源的信息往往较少，多数情况下该来源值为空。

（4）从正文内容中获取文档标题。在正常的富文档中，正文顶部最大字体且无结束句号的单独成段文字往往可能是文档标题，可以分别利用 Apache PDFBox 和 Apache POI 解析获取其首页的内容并从首页内容中按行取出同一字体的最大字体内容。

针对此解决的方法主要如图 5 所示。笔者主要是取了三处信息，包括富文档第一页上最大字体内容和第二大字体内容，同时采用了富文档的来源 url 中相关信息作为辅助判定信息。通过这种方法，笔者可以准确识别出大约 80%以上的富文档文件的标题信息。

```
Delcare Text as rich Document’s Content;
Declare richTitle as  rich Document’s  final Title;
Declare CandateTitle1;
Declare CandateTitle2;
Declare CandateTitle3;
解析出第一页的内容信息（包括格式信息）
CandateTitle1=第一页内容信息中的最大字体内容；
CandateTitle2=第一页内容信息中的第二号字体内容；
CandateTitle3=该富文档url中最后一个“/”至文档格式符如“.pdf”前的内容；
if（CandateTitle1.token.size> 设定阈值）
{
  if(CandateTitle1中包含了“。”、“？”、“！”号等句子结束符){
  截断Candatetitle1至这些符号；
  }else{
      替换candateTitle1中所有的格式符
  }
  richTitle=candateTitle1;
}else{
  将candateTitle1和candateTitle2拼接在一起，重复上面的步骤；
   若该拼接字符还不符合要求，则再取用candateTitle3的字符内容。
}
```

图 5 获取 pdf 文档 title 的方案流程

（3）　正文内容的提取

以 PDF 文档为例，针对 PDF 类型文档的正文内容的提取，笔者主要是利用了开源工具 PDFBox 对此类文档进行解析抽取。笔者采用的是 PDFBox-1.6 版本。PDFBox 是一个开源的 Java 文档 API 库，通过这个库可以访问 PDF 文件的各项信息，还可以进行创建、处理以及文档内容提取等一系列操作。笔者抽取 PDF 文档正文内容主要方法流程如图 6 所示：

```
FileOutputStream fstream = null;
BufferedOutputStream stream = null;
File file = null;
PDDocument document = null;
String result = "";
int startPage = 1;
int endPage = Integer.MAX_VALUE;
try {
    InputStream input = new ByteArrayInputStream(htmlfile);
    document = PDDocument.load(input);
    PDFTextStripper stripper = new PDFTextStripper();
    stripper.setSortByPosition(true);
    stripper.setStartPage(startPage);
    stripper.setEndPage(endPage);
    result = stripper.getText(document);
    System.out.println("---read pdf success!---");
} catch (Exception e) {
    System.out.println("---read pdf failed!---");
} finally {
    if (document != null) {
        try {
            document.close();
        } catch (IOException e) {
            System.out.println("---close pdf failed!---");
        } finally {
            document = null;
            System.gc();
```

图 6 PDFBox 提取正文内容的主要方案流程

通过 PDFBox 能够解析大多数 PDF 文档，但是笔者通过观察实际的 PDF 采集过程中发现，有部分 PDF 文档未能解析出正文内容，而这些 PDF 文档对科研人员也有很有价值，为解决科研用户的这种需求，笔者调研了目前一些其他开源的文本解析工具，通过调查、测试发现 Tika 很好的解决此类问题。Tika[10]是 Apache 的 Lucene 项目下面的子项目，它集成了现有的不同类型的文档解析库，并提供统一的接口，便于操作不同类型的富文档。通过 Tika 可以自动检测各种富文档的类型并抽取这些不同类型的富文档的元数据及正文内容信息。笔者采用 Tika 解析 PDF 文档正文内容的主要方法流程如图 7 所示：

```
Tika tika = new Tika();
tika.setMaxStringLength(1000*1000);
Metadata metadata = new Metadata();
metadata.set(Metadata.AUTHOR, "空号");
metadata.set(Metadata.RESOURCE_NAME_KEY, "pdf");
URL url = new URL(u);
String str = tika.parseToString(url);
for (String name : metadata.names()) {
    System.out.println(name + ":" + metadata.get(name));
}
return str;
```

图 7 Tika 提取正文内容的主要方案流程

综上所述，针对 PDF 类型的富文档正文内容的提取，笔者首先采用 PDFBox

进行提取，如果提取出的正文内容为空且为 PDF 文档，接着采用 Tika 进行提取。

4 富文档监测与识别的应用效果

基于上述的富文档监测和识别框架集方法，笔者综合利用了 htmlParser[7]、PDFBox[8]、POI[9]、Tika[10]工具等实现了富文档载体类型判定和元数据获取的相关小组件，并将这些小组件应用于实际的网络监测系统中，取得了较好的效果。目前，笔者所在的项目组共针对 6578 个监测信息源、18723 个监测目录开展常规的动态监测，共采集资源量达到了 10750006 条，其中，基于本文提出的富文档监测与识别方法，共判定出 310528 个 PPT、PDF、DOC、XLS 类型的富文档文件，并准确获得了这些文件的标题和发布时间，如图 6 所示。笔者在实际应用中，通过随机抽取 2000 篇富文档进行分析，通过比较系统识别的标题、发布时间等元数据和人工识别的标题、发布时间等元数据，从中筛选出比较吻合的记录，从而得出其识别准确率。在实际比对过程中，如果识别的元数据与人工识别的元数据出现了细微的偏差，如标题不是太完整，发布时间年月吻合等，笔者将这样的记录也认为是基本吻合的。具体的测试结果如表所示。这些解析出的标题和发布时间在后续的基于题名的文本聚类以及时间线上的趋势变化分析中有着重要的基础支撑作用。

The Effectiveness of Mindfulness Based Interventions i...	pdf	2016-05-18 21:24:25.000	http://sro.sussex.ac.uk/61180/1/__smbhome.uscs.s...
No load shedding during Sahr, Iftaar: Abid Sher	pdf	2016-05-28 02:52:53.000	http://www.brecorder.com/top-news/pakistan/29834...

图 6 判定出的富文档类型、标题和发布时间示例

资源	样本总数	吻合记录数	识别准确率
富文档文件	2000 条	1759 条	87.95%

表 2 富文档元数据识别效率测试结果

5 总结

通过对富文档监测与识别的研究与探索，笔者拓展了文本知识内容的识别方法，为后续的深度知识分析提供了有效的支撑。但是在实际的应用中笔者也发现现有的解决方案中无法解决的问题，比如，还有极少数将 html 文件判定为 pdf 文件的情况出现，识别出的标题中包含了较多的乱字符。通过在今后不断的语料累积中，笔者可以进一步发现这些错误情况的规律，进而提出相应的方法解决。

参考文献：

[1]Open Document Format for Office Applications (OpenDoeument) v1.0 [S]. http://docs.oasis-open.or~oficdv1.0.

[2] Laurette P Simmon. 文件中色彩、字形和字体变化的显著突出作用[J]. 计算机工程，2000，26(11).

[3]Mao, S., Rosenfeld, A., & Kanungo, T. (2003). Document structure analysis algorithms: a literature survey. Proc. SPIE Electronic Imaging, (pp. 197 - 207).

[4] Fujiyoshi, A., Suzuki, M., & Uchida, S. (2009). Syntactic Detection and Correction of Misrecognitions in Mathematical OCR. ICDAR, (pp. 1360 - 1364).
[5] Minh-Thang Luong, Thuy Dung Nguyen, and Min-Yen Kan. to appear. Logical structure recovery in scholarly articles with rich document features. International Journal of Digital Library Systems, 1(4):1-23, Jan. 2011
[6] Multipurpose Internet Mail Extensions-(MIME) Part Two: Media Types. http://www.mhonarc.org/~ehood/MIME/2046/rfc2046.html
[7] HTML Parser. http://htmlparser.sourceforge.net/
[8] Apache PDFBox. http://pdfbox.apache.org/
[9] Apache POI. http://poi.apache.org/
[10]Apache Tika. https://tika.apache.org/
[11]李珍，田学东.PDF文件信息的抽取与分析[J].计算机应用，2003，23(12)
[12]张秀秀，马建霞.PDF科技论文语义元数据的自动抽取研究[J].现代图书情报技术，2008,11
[13]牛永洁，薛苏琴.基于PDFBox抽取学术论文信息的实现[J].计算机技术与发展，2014,24(12)

作者简介：
张敏，1985 年生，助理馆员，硕士，主要研究方向：信息采集，E-mail:zhangmin@mail.las.ac.cn
刘建华，1984 年生，情报学博士在读，馆员，主要研究方向：信息抽取、文本挖掘、网络计量等，E-mail：liujh@mail.las.ac.cn
谢靖，1983 年生，馆员，硕士，主要研究方向：语义索引、智能信息处理，E-mail:xiej@mail.las.ac.cn